



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2002-0075192
Application Number PATENT-2002-0075192

출원 년 월 일 : 2002년 11월 29일
Date of Application NOV 29, 2002

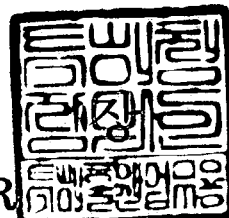
출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Institute



2002 년 12 월 18 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0005
【제출일자】	2002.11.29
【발명의 명칭】	파일 레벨 스트라이핑 장치 및 방법
【발명의 영문명칭】	APPARATUS AND METHOD FOR FILE-LEVEL STRIPING
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	장성구
【대리인코드】	9-1998-000514-8
【포괄위임등록번호】	2001-038646-2
【대리인】	
【성명】	김원준
【대리인코드】	9-1998-000104-8
【포괄위임등록번호】	2001-038648-7
【발명자】	
【성명의 국문표기】	김창수
【성명의 영문표기】	KIM, Chang-soo
【주민등록번호】	691206-1533317
【우편번호】	305-762
【주소】	대전광역시 유성구 전민동 464-1 엑스포아파트 410-704
【국적】	KR
【발명자】	
【성명의 국문표기】	박유현
【성명의 영문표기】	BAK, Yuhyeon
【주민등록번호】	731123-1024522
【우편번호】	302-766
【주소】	대전광역시 서구 탄방동 산호아파트 102-409
【국적】	KR

【발명자】

【성명의 국문표기】 김영호
【성명의 영문표기】 KIM, Young Ho
【주민등록번호】 731210-1382411
【우편번호】 305-350
【주소】 대전광역시 유성구 가정동 236-1 (16/5) 신관-118
【국적】 KR

【발명자】

【성명의 국문표기】 강동재
【성명의 영문표기】 KANG, Dong Jae
【주민등록번호】 721213-1067419
【우편번호】 302-723
【주소】 대전광역시 서구 관저동 1144 구봉마을아파트 708-103
【국적】 KR

【발명자】

【성명의 국문표기】 김학영
【성명의 영문표기】 KIM, Hag Young
【주민등록번호】 610705-1804411
【우편번호】 305-330
【주소】 대전광역시 유성구 지족동 열매마을아파트 503-1002
【국적】 KR

【발명자】

【성명의 국문표기】 김명준
【성명의 영문표기】 KIM, Myung-Joon
【주민등록번호】 550807-1024619
【우편번호】 305-340
【주소】 대전광역시 유성구 도룡동 393-17
【국적】 KR

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
장성구 (인) 대리인
김원준 (인)

【수수료】

【기본출원료】 20 면 29,000 원

【가산출원료】 9 면 9,000 원

【우선권주장료】 0 건 0 원

【심사청구료】 10 항 429,000 원

【합계】 467,000 원

【감면사유】 정부출연연구기관

【감면후 수수료】 233,500 원

【기술이전】

【기술양도】 희망

【실시권 허여】 희망

【기술지도】 희망

【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 파일 레벨 스트라이핑 장치 및 방법에 관한 것으로, 물리 블록 넘버 (physical block number)를 통해 액세스되며, 정보를 실제로 저장하는 다수의 디스크; 다수의 디스크를 논리적으로 연결하여 하나의 커다란 논리 볼륨을 형성하고, 논리 볼륨을 관리하기 위해 필요한 정보를 다수의 디스크에 기록하고 관리하는 볼륨 관리자; 볼륨 관리자를 통해 제공되며, 논리 블록 넘버(logical block number)를 통해 액세스되는 논리 볼륨; 볼륨 관리자에 의해 제공되는 논리 볼륨을 하나의 저장 장치로 인식하며, 제공된 논리 볼륨 상에 파일들을 생성하고, 생성된 파일에 대한 입/출력을 수행하기 위해 논리 볼륨에 대하여 논리 블록 넘버를 적용하는 파일 시스템을 구비한다. 또한, 파일 생성 인터페이스에 파일 레벨 스트라이핑 여부를 지시하는 옵션을 추가하고, 아이노드 구조에 해당 파일을 위해 마지막으로 물리 블록 할당이 발생한 디스크 ID를 기록하기 위한 필드를 추가하여 이를 기반으로 볼륨 관리자가 물리 블록 할당을 수행함으로써, 전체 논리 볼륨뿐만 아니라 각 파일을 위한 데이터 블록들이 논리 볼륨에 참여하는 전체 디스크에 균등하게 분배되어 대용량 파일에 대한 입/출력의 성능을 향상시킬 수 있는 효과가 있다.

【대표도】

도 4

【명세서】

【발명의 명칭】

파일 레벨 스트라이핑 장치 및 방법{APPARATUS AND METHOD FOR FILE-LEVEL STRIPING}

【도면의 간단한 설명】

도 1은 종래 볼륨 관리자의 스트라이핑 기법에 대하여 도시한 도면이고,

도 2는 종래 아이 노드(i node) 구조와 파일 데이터의 디스크 분포에 대하여 도시한 도면이며,

도 3은 종래 파일 데이터의 근접 할당 예에 대하여 도시한 도면이며,

도 4는 본 발명에 따른 파일 레벨 스트라이핑 장치에 대한 도면이며,

도 5는 본 발명에 따른 볼륨 관리자에 의해 관리되는 디스크의 내부 구조를 도시한 도면이며,

도 6은 본 발명에 따른 아이노드 구조와 특정 파일을 위한 블록 할당 예에 대하여 도시한 도면이며,

도 7은 본 발명에 따른 파일 시스템에서의 파일 생성 시, Last Disk ID 값을 초기화하는 과정에 대하여 상세하게 설명한 도면이며,

도 8은 본 발명에 따른 볼륨 관리자에서의 물리 블록 할당 및 파일 시스템에서의 Last Disk ID 값 변경 절차에 대하여 상세하게 설명한 도면이며,

도 9는 도 6에 도시된 실시 예에 따른 파일 입/출력 절차에 대하여 상세하게 설명한 도면이다.

<도면의 주요부분에 대한 부호의 설명>

100 : 파일 시스템

200 : 논리 볼륨

300 : 볼륨 관리자

400 : 디스크

501 : 볼륨 레이블

502 : 자유공간 비트맵

503 : 매핑 테이블

504 : 물리 블록 또는 익스텐트(Extent)

505 : 볼륨 헤더

506 : 데이터 영역

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<16> 본 발명은 파일 레벨 스트라이핑 장치 및 방법에 관한 것으로, 특히 파일 레벨 스트라이핑 기법을 이용하여 볼륨 관리자(Volume Manager)와 파일 시스템(File System)에서 2개 이상의 디스크(Disk)로 각 파일 데이터(Data)를 분산 저장하도록 하는 장치 및 방법에 관한 것이다.

<17> 통상적으로, 컴퓨터 환경은 인터넷(Internet)의 발전으로 인하여 동영상 서비스 등의 멀티미디어(Multimedia) 기반의 서비스가 주를 이루고 있으며, 이 서비스를 제공하기 위해 관리하는 파일의 크기가 대형화되는 것이 현 실정이다.

<18> 즉, 대용량 데이터에 대한 고속의 서비스를 제공하기 위해 데이터에 대한 고속의 입/출력 기능이 필수적으로 요구되는 것으로, 이 고속의 서비스를 제공하기 위해 볼륨 관리자나 레이드(Redundant Array of Inexpensive Disk : RAID) 시스템에서는 스트라이핑 기법을 적용한다.

- <19> 이러한, 스트라이핑 기법은 데이터를 2개 이상의 디스크에 분산 저장한 후, 입/출력을 병렬 방식으로 수행하여 그 성능을 향상시킬 수 있는 기법이다.
- <20> 또한, 파일 시스템 분야에서는 파일 시스템이 단일 디스크 상에 존재(예로, 볼륨 관리자나 RAID 시스템을 사용할 경우, 2개 이상의 디스크를 하나의 논리적인 장치로 인식시키므로 파일 시스템에서는 하나의 장치로 인식한다)하고, 각 파일을 위한 데이터들이 디스크 상에 흩어져 존재할 수 있으므로, 이들 데이터 조각들을 어떻게 근접하게 위치시킬 것인가에 초점을 맞추면서, 파일 시스템에서 입/출력의 기본 단위인 블록의 크기를 크게 설정하여 성능 향상을 기대하였다.
- <21> 그러나, 볼륨 관리자나 RAID 시스템에 있어서의 스트라이핑 기법은 도 1의 종래 볼륨 관리자의 스트라이핑 기법에 대하여 도시한 도면의 예처럼, 데이터를 논리 주소를 기준으로 분배한다.
- <22> 즉, 논리 주소 상 첫 번째 블록(Block)인 블록 0(110)은 디스크 0의 첫 번째 블록(120)에 대응되며, 논리 주소 두 번째 블록은 디스크 1의 첫 번째 블록에 대응되도록 각 논리주소의 블록을 각 디스크의 블록에 순차적으로 배치한다. 이와 더불어, 대용량 파일을 위한 입/출력 성능 향상을 위해 블록은 여러 개의 연속된 블록을 합친 익스텐트(Extent)의 개념으로 확장시킬 수 있다.
- <23> 다음으로, 도 2는 종래 아이 노드(i node) 구조와 파일 데이터의 디스크 분포에 대하여 도시한 도면으로서, 이중 아이노드 구조는 해당 파일에 대한 정보를 저장하는 헤더 영역(210)과, 해당 파일의 데이터를 저장하는 데이터 블록들을 지시하기 위한 다수개의 데이터 블록에 대한 포인터(220)들로 구성된다.

- <24> 즉, 도 2를 참조하면, 파일 시스템에서 각 파일의 데이터 저장은 연속적인 논리 블록을 할당하지 않고 전체 디스크 또는 논리 볼륨에 흩어져 존재한다. 이에 따라, 특정 파일을 위한 데이터가 전체 디스크에 균등하게 분배되지 않고 특정 디스크에 집중될 가능성이 존재하며, 이 경우 해당 파일을 위한 입/출력 연산은 효과적인 성능을 제공하지 못하게 되는 문제점이 있다.
- <25> 또한, 파일 시스템에서 멀티미디어와 같은 대용량 파일에 대한 효율적인 입/출력을 제공하기 위해 도입한 데이터 블록의 근접 할당은 도 3에 도시된 종래 파일 데이터의 근접 할당 예에 대하여 도시한 도면에서 보는 것처럼, 블록 대신 익스텐트(Extent)를 사용하는 것과 동일한 효과를 가져오며 익스텐트의 크기가 변경 가능한 구조이며, 이 외에 블록의 크기를 크게 하는 방법 역시 블록 대신 익스텐트를 사용하는 것과 동일하며, 이들은 일반 파일 시스템과 동일하게 특정 파일을 위한 데이터가 전체 디스크에 균등하게 분배되지는 않아 대용량 파일에 대한 입/출력 연산을 효율적으로 처리하지 못하게 되는 문제점이 있다.
- <26> 이와 같이, 파일 레벨 스트라이핑 기법과 관련하여, 선행기술로는 1997년 12월 23일로 10-1997-0072755로 등록된 "레이드 레벨 5 시스템에서의 빠른 시스템 재구성 방법"과, 1998년 10월 27일 US 5,828,876으로 등록된 "File system for clustered processing system"와, 2001년 11월 6-10일 2001 Annual Linux Showcase/Usenix에 게재된 "A Persistent Snapshot Device Driver for Linux" 등에 개시되어 있다.
- <27> 상술한 바와 같이, 개시된 선행기술을 상세하게 설명하면, 레이드 레벨 5 시스템에서의 빠른 시스템 재구성 방법은 시스템을 재구성한다는 것은 전체 디스크로 분산 저장되어 있는 데이터 및 패리티 블록들을 새롭게 배치하는 것으로, 종래에는 재구성을 위해

시스템의 수행을 중단시킨 후, 전체 디스크의 내용을 읽어서 배치 방식에 따라 다시 디스크로 쓰는 과정으로 처리되었다.

<28> 따라서, 해당 디스크의 내용을 임시로 저장할 메모리에 드는 비용과 블록들의 재배치를 위한 여러 번의 디스크 읽기 및 쓰기 연산의 수행에 걸리는 시간 때문에 재구성 과정은 시스템 성능에 커다란 오버헤드가 되었다.

<29> 그러나, 이 방식은 시스템의 수행을 중단할 필요 없이 각 디스크의 로드(load)에 따라 스트립(stripe) 단위로 재구성 과정을 수행하며, 각 과정은 한번의 디스크 읽기 및 쓰기를 위한 단 2번의 디스크 연산으로 처리되기 때문에 시간을 단축시킬 수 있으며, 또한 시스템 전체를 중단할 필요가 없기 때문에 시스템 성능에 미치는 영향도 적은 시스템 재구성 방법을 제안하는 기술이다.

<30> 다음으로, File system for clustered processing system은 프로세싱 노드들과 이들 노드를 연결하는 연결 네트워크를 포함하는 유닉스 클러스터 컴퓨터 시스템에서 데이터를 저장하고 이를 검색하는 효과적인 파일 시스템이다.

<31> 즉, 향상된 파일 시스템은 하나의 프로세싱 노드에 직접 붙어있는 장치보다는 각 프로세싱 노드에 공유 스커시 장치 형태로 연결되어 있는 디스크장치 같은 데이터 저장장치이다. 데이터 저장장치를 관리하기 위한 파일 시스템의 전체 구조는 각 프로세싱 노드가 저장장치를 접근하기 위한 모든 정보를 포함하고 있다.

<32> 이러한, 파일 시스템은 파일 시스템을 관리하기 위한 슈퍼블록, 아이노드 비트맵, 수정된 저널영역, 아이노드 영역, 데이터 블록 비트맵, 그리고 데이터 블록 영역으로 구

분되며, 파일 시스템은 시스템의 이용을 조정하기 위한 분산 잠금 관리자와의 인터페이스를 이용한다.

<33> 다음으로, A Persistent Snapshot Device Driver for Linux은 웹서버나 대형 엔터프라이즈 시스템에서 요청되는 데이터의 연속적인 가용성을 제공하는 온라인 백업을 제공한다. 또한 기존의 오프 라인 백업의 수행을 위한 시스템의 중단을 없앨 수 있으며, 온라인 백업에 대한 요구는 오프 라인의 백업 수행을 위해 서버가 중단됨으로써 발생하는 생산성의 손실이 대용량으로 갈수록 증가된다.

<34> 스냅샷은 온라인 백업을 수행할 수 있도록 지원하는 것으로, 이 논문에서는 클러스터 환경에서의 영속적인 스냅샷을 제공하는 리눅스 기반의 디바이스 드라이버를 제공하며, 스냅샷의 영속성을 제공하기 위해 매핑과 변경 블록과 같은 메타 데이터들을 로그 디스크에 기록하며, 로그의 디스크 반영 중 발생할 수 있는 시스템 에러를 대비하여 트랜잭션 식별자를 도입하며, 클러스터 환경의 지원을 위해 매핑 정보 같은 메타 데이터 접근을 직렬화하기 위한 전역 잠금 관리자를 제공하는 기술이다.

<35> 이와 같이, 선행 특허에 개시된 기술을 살펴보았을 때, 본원 발명에서의 문제점인 특정 파일을 위한 데이터가 전체 디스크에 균등하게 분배되지 않고 특정 디스크에 집중될 가능성이 존재하여, 대용량 파일에 대한 입/출력 연산을 효율적으로 처리하지 못하여 효과적인 성능을 제공하지 못하게 되는 문제점을 여전히 갖고 있다.

【발명이 이루고자 하는 기술적 과제】

<36> 따라서, 본 발명은 상술한 문제점을 해결하기 위해 안출된 것으로서, 그 목적은 파일 레벨 스트라이핑 기법을 이용하여 볼륨 관리자(Volume Manager)와 파일 시스템(File

System)에서 2개 이상의 디스크(Disk)로 각 파일 데이터(Data)를 분산 저장하도록 하는 파일 레벨 스트라이핑 장치 및 방법을 제공함에 있다.

<37> 상술한 목적을 달성하기 위한 본 발명에서 파일 레벨 스트라이핑 장치는 물리 블록 넘버(physical block number)를 통해 액세스 되며, 정보를 실제적으로 저장하는 다수의 디스크; 다수의 디스크를 논리적으로 연결하여 하나의 커다란 논리 볼륨을 형성하고, 논리 볼륨을 관리하기 위해 필요한 정보를 다수의 디스크에 기록하고 관리하는 볼륨 관리자; 볼륨 관리자에 의해 제공되는 논리 볼륨을 하나의 저장 장치(storage device)로 인식하며, 제공된 논리 볼륨 상에 파일을 생성한 후, 생성된 파일에 대한 입/출력을 수행하기 위해 논리 볼륨에 대하여 논리 블록 넘버를 적용하는 파일 시스템을 포함하는 것을 특징으로 한다.

<38> 또한, 상술한 목적을 달성하기 위한 본 발명에서 파일 시스템과 볼륨 관리자를 구비한 파일 레벨 스트라이핑 기법은 파일 시스템에서 파일을 생성하는 인터페이스(interface)에 파일 레벨 스트라이핑 지원 여부를 지시하는 옵션(option)을 추가하는 단계; 파일을 위한 아이노드 구조에 Last Disk ID를 확장하는 단계; 파일 시스템에서 파일을 생성할 때 Last Disk ID를 초기화 하는 단계; 파일 시스템에서 파일에 대한 입/출력 요청 시, 볼륨 관리자에 의해 물리 블록 할당이 요구될 때, Last Disk ID를 기반으로 물리 블록을 할당하는 단계; 볼륨 관리자에 의해 물리 블록 할당이 발생할 때, 파일 시스템에서 이것을 반영하기 위해 Last Disk ID 값을 변경하는 단계를 포함하는 것을 특징으로 한다.

【발명의 구성 및 작용】

- <39> 이하, 첨부된 도면을 참조하여 본 발명에 따른 실시 예를 상세하게 설명하기로 한다.
- <40> 도 4는 본 발명에 따른 파일 레벨 스트라이핑 장치에 대한 도면으로서, 파일 시스템(100)과, 논리 볼륨(200)과, 볼륨 관리자(300)와, 다수의 디스크(400,...)를 포함한다.
- <41> 파일 시스템(100)은 볼륨 관리자(300)에 의해 제공되는 논리 볼륨(200)을 하나의 저장 장치(storage device)로 인식하며, 제공된 논리 볼륨(200) 상에 파일들을 생성하고, 생성된 파일에 대한 입/출력을 수행하기 위해 논리 볼륨(200)에 대하여 논리 블록 넘버를 적용한다.
- <42> 논리 볼륨(200)은 볼륨 관리자(300)에 의해 다수의 디스크(400,...)가 논리적으로 연결되어 하나의 가상적인 저장장치로 인식되도록 제공되며, 논리 블록 넘버를 통해 액세스 된다.
- <43> 볼륨 관리자(300)는 일반적인 컴퓨터 시스템이나 레이드 시스템 내에서 동작하는 블록으로서, 사용자에게 의해 지정된 다수의 디스크(400,...)를 논리적으로 연결하여 하나의 커다란 논리 볼륨(200)을 형성하고, 논리 볼륨(200)을 관리하기 위해 필요한 정보를 다수의 디스크(400,...)에 기록하고 관리한다.
- <44> 디스크(400,...)는 물리 블록 넘버를 통해 액세스 되며, 정보가 실제로 저장되는 매체이다.

- <45> 도 5는 본 발명에 따른 볼륨 관리자(300)에 의해 관리되는 디스크의 내부 구조를 도시한 도면으로서, 보다 상세하게는 도 4에 도시된 임의의 디스크(400)를 확대한 도면이다.
- <46> 즉, 볼륨 관리자(300)는 논리 볼륨(200)을 관리하기 위해 필요한 정보를 디스크(400)에 저장하기 위한 영역인 볼륨 레이블(Volume Label)(501)을 생성한다.
- <47> 이후, 디스크(400)에 대한 물리 블록 할당을 제어하기 위한 자유공간 비트맵(Free Space Bitmap)(502)을 지정하여 유지 관리하며, 특정 논리 블록이 어떤 물리 블록(504)에 대응되는지에 대한 정보로서 매핑 테이블(Mapping Table)(503)을 유지한다.
- <48> 다시 말해서, 디스크(400)내에 볼륨 레이블(501), 자유공간 비트맵(502), 매핑 테이블(503) 정보 등의 모임인 볼륨 헤더(Volume Header)(505)를 제외한 나머지 공간에 데이터를 저장할 물리 블록들의 모임인 데이터 공간(Data Area)(506)으로 설정하는 방식으로 관리한다.
- <49> 한편, 본 발명에 따른 파일 레벨 스트라이핑 기법을 가능하게 하기 위한 구조 확장은, 도 4의 파일 시스템(100)에서 파일을 생성하는 인터페이스(interface)에 파일 레벨 스트라이핑 지원 여부를 지시하는 옵션(option)을 추가하는 것과, 도 6의 아이노드 구조에서 해당 파일을 위해 마지막으로 물리 블록 할당이 발생된 디스크(400,...)의 아이디(ID)에 대한 정보를 저장하기 위한 필드인 Last Disk ID(601)를 추가하는 것을 포함한다.
- <50> 도 7은 파일 시스템(100)에서 파일 생성 시, 도 6의 확장된 아이노드 구조 내의 Last Disk ID(601)를 초기화하는 과정에 대하여 상세하게 설명한다.

- <51> 즉, 도 4의 파일 시스템(100)은 파일을 생성하는 인터페이스에서 옵션으로 주어진 모드(mode)에서 파일 레벨 스트라이핑을 지정하는 비트가 세팅되어 있는지를 판단한다(단계 701).
- <52> 상기 판단 단계(701)에서 해당 비트가 세팅되어 있지 않으면, 해당 파일이 파일 레벨 스트라이핑을 지원하지 않는다는 의미로서, Last Disk ID(601) 값을 "-1"로 초기화한다(단계 702).
- <53> 반면에, 상기 판단 단계(701)에서 해당 비트가 세팅되어 있으면, 해당 파일이 파일 레벨 스트라이핑을 지원한다는 의미로서, Last Disk ID(601) 값을 "0~(논리 볼륨에 참여하는 디스크 수 - 1)"의 범위에 있는 정수 중 무작위 정수를 선택하여 초기화한다(단계 703).
- <54> 여기서, 무작위 정수를 선택하는 이유는 파일 레벨 스트라이핑을 지원하는 모든 파일의 초기값이 특정 값으로 고정될 경우, 해당 값에 대응되는 디스크에 많은 데이터 블록의 할당이 집중될 가능성을 제거하기 위한 것이다.
- <55> 한편, 도 8은 도 6을 참조하면서, 본 발명에 따른 파일 레벨 스트라이핑 기법을 적용할 경우, 파일 시스템(100)에 의하여 임의의 파일이 생성되고, 해당 파일을 위한 아이노드 내의 Last Disk ID(601) 값이 초기화된 이후, 해당 파일에 대한 입/출력이 진행되는 과정을 상세하게 설명한다.
- <56> 파일 시스템 사용자는 상술한 파일에 대해 파일 오프셋(offset) 정보를 제공하며 입/출력을 요청한다(단계 801).

- <57> 파일에 대해 입/출력이 요청되면, 파일 시스템은 해당 파일을 위한 아이노드에서 파일 오프셋에 대응되는 논리 블록 포인터(602)를 검사하여 파일 입/출력에 대응되는 논리 블록을 결정한다. 만약 논리 블록 포인터가 널(null) 값이면, 해당 파일 오프셋에 할당된 논리 블록이 존재하지 않는 것이므로, 사용되지 않는 논리 블록들 중 하나를 선택하여 해당 파일 오프셋에 할당하고, 할당 결과를 아이노드의 논리 블록 포인터에 기록한다(단계 802).
- <58> 상기 논리 블록 결정 단계(802)가 완료되면, 파일 시스템은 결정된 논리 블록에 대한 파일 입/출력을 하위 입/출력 시스템인 볼륨 관리자에 요청한다(단계 803).
- <59> 이후, 볼륨 관리자(300)는 해당 논리 블록에 대응되는 디스크와, 그 디스크 내의 물리 블록을 결정하기 위하여 주소 매핑 과정을 수행한다(단계 804).
- <60> 주소 매핑 과정을 수행한 결과, 해당 논리 블록에 대응되는 디스크 및 물리 블록이 결정되어 물리 블록 할당이 불필요한지, 아니면 해당 논리 블록이 처음 사용되어 해당 논리 블록을 위한 물리 블록의 할당이 필요한지를 체크한다(단계 805).
- <61> 상기 체크 단계(805)에서 물리 블록 할당이 필요하지 않은 경우, 결정된 디스크 및 물리 블록에 대하여 입/출력을 수행하는 단계(810)를 진행한다.
- <62> 반면에, 물리 블록 할당의 필요성 체크 단계(805)에서 물리 블록 할당이 요구되는 경우, 볼륨 관리자(300)는 해당 파일에 대응되는 아이노드의 Last Disk ID(601) 값을 체크한다(단계 806).
- <63> 상기 체크 단계(806)에서 Last Disk ID(601) 값이 "-1"이면, 해당 파일이 파일 레벨 스트라이핑을 수행하지 않는다는 의미이므로 기존의 스트라이핑 기법에서와 동일하게

전체 볼륨을 대상으로 다음 물리 블록 할당을 수행해야 할 디스크를 결정하기 위한 변수를 참조하여 물리 블록 할당을 수행할 디스크를 선택한다(단계 807).

<64> 반면에, 상기 체크 단계(806)에서 Last Disk ID(601) 값이 "0~(논리 볼륨에 참여하는 디스크 수 - 1)"의 범위이면, 해당 파일이 파일 레벨 스트라이핑을 지원한다는 의미로서, Last Disk ID(601) 값의 다음 값을 가지는 디스크를 선택한다(단계 808).

<65> 볼륨 관리자(300)는 물리 블록 할당을 수행할 디스크가 결정되면, 결정된 디스크의 자유공간 비트맵(502)을 참조하여 물리 블록 할당을 수행하고 물리 블록 할당 결과를 매핑 테이블(503)에 반영한다(단계 809).

<66> 이후, 볼륨 관리자(300)는 주소 매핑 과정 수행 단계(804) 이후의 단계들(805, 806, 807, 808, 809)을 수행하면서 입/출력을 수행할 디스크 및 그 디스크 내의 물리 블록을 결정하면, 해당 물리 블록에 입/출력을 수행한다(단계 810).

<67> 물리 블록으로의 입/출력이 완료되면, 볼륨 관리자(300)는 파일 시스템(100)에 물리 블록 할당 발생 여부와 물리 블록 할당이 발생한 디스크 정보를 통보하고(단계 811), 파일 시스템(100)은 물리 블록 할당이 발생하였고 Last Disk ID(601) 값이 -1이 아닌지를 검사한다(단계 812).

<68> 상기 검사 단계(812)에서 물리 블록 할당이 발생하였고 Last Disk ID(601) 값이 -1이 아니면, 파일 시스템(100)은 해당 파일을 위한 아이노드 내 Last Disk ID(601) 값을 볼륨 관리자(300)가 통보한 물리 블록 할당이 발생한 디스크 ID 값으로 변경하고, 파일 입/출력 연산을 종료한다(단계 813).

- <69> 한편, 도 9는 도 6의 실시 예에 따른 입/출력 흐름도로서, 본 발명의 일 실시 예에 따른 파일 레벨 스트라이핑 기법에 대하여 설명한다.
- <70> 먼저, 파일 시스템(100)에 의해 파일 생성 시, 주어진 옵션에 따라 파일 레벨 스트라이핑의 지원여부가 결정되는데, 파일 레벨 스트라이핑을 지원하는 경우라 가정한다.
- <71> 즉, 해당 파일을 위한 아이노드 내 Last Disk ID(601) 값은 도 6에 도시된 바와 같이, 3개의 디스크로 이루어져 있으므로, 디스크0, 디스크1, 디스크2 중 하나의 값으로 결정되는데, 본 발명의 실시 예에서는 Last Disk ID(601)의 초기값을 "0"으로 세팅하였다고 가정한다.
- <72> 이후, 파일에 대한 데이터 입/출력이 요청될 경우(단계 901), 파일 시스템(100)은 논리 블록 1을 할당하고 아이노드에 할당된 값을 반영(603)한다(단계 902).
- <73> 그리고, 볼륨 관리자(300)에게 논리 블록 1에 대한 입/출력을 요청하면, 논리 블록 1이 처음 사용되는 경우이므로 Last Disk ID(601) 값을 기반으로 물리 블록을 할당한다(단계 903). 즉, Last Disk ID(601) 값이 "0"이므로 물리 블록 할당은 디스크 1번에서 수행되며, 그 결과로서 디스크 1의 세 번째 블록이 할당(603-1)된다.
- <74> 볼륨 관리자(300)에 의해 할당되었다는 결과가 매핑 테이블에 기록되며, 디스크 1의 세 번째 블록에 대하여 입/출력이 수행된다(단계 904).
- <75> 입/출력이 완료된 후, 볼륨 관리자(300)는 Last Disk ID(601)의 변경 필요성을 알려주며, 파일 시스템(100)은 이를 반영하기 위하여 Last Disk ID(601) 값을 "1"로 변경한다(단계 905).

- <76> 이후, 논리 블록 1의 영역이 모두 사용된 후, 파일에 대한 입/출력 요청이 발생하면 새로운 논리 블록 5의 할당이 이루어지며 이 결과가 아이노드에 반영(604)된다(단계 906).
- <77> 논리 블록 5에 대한 입/출력 요청은 새로운 물리 블록 할당을 요구하게 되는 것으로, 볼륨 관리자(300)는 현재의 Last Disk ID 값이 "1"인 점에 착안하여 물리 블록 할당을 디스크 2의 세 번째 블록에 대하여 수행(604-1)하고 입/출력을 수행하는 것이다(단계 907).
- <78> 입/출력 수행이 완료된 후, 볼륨 관리자(300)는 Last Disk ID(601)의 변경 필요성을 알려주며, 파일 시스템(100)은 이를 반영하기 위하여 Last Disk ID(601) 값을 "2"로 변경한다(단계 908).
- <79> 이후, 논리 블록 5의 영역이 모두 사용된 후, 파일에 대한 입/출력 요청이 발생하면 새로운 논리 블록 7의 할당이 이루어지며 이 결과가 아이노드에 반영(605)된다(단계 909).
- <80> 논리 블록 7에 대한 입/출력 요청은 새로운 물리 블록 할당을 요구하게 되는 것으로, 볼륨 관리자(300)는 현재의 Last Disk ID 값이 "2"인 점에 착안하여 물리 블록 할당을 디스크 0의 네 번째 블록에 대하여 수행(605-1)하고 입/출력을 수행하는 것이다(단계 910).
- <81> 입/출력 수행이 완료된 후, 볼륨 관리자(300)는 Last Disk ID(601)의 변경 필요성을 알려주며, 파일 시스템(100)은 이를 반영하기 위하여 Last Disk ID(601) 값을 "0"으로 변경한다(단계 911).

<82> 이후, 논리 블록 7의 영역이 모두 사용된 후, 파일에 대한 입/출력 요청이 발생하면 새로운 논리 블록 11의 할당이 이루어지며 이 결과가 아이노드에 반영(606)된다(단계 912).

<83> 논리 블록 11에 대한 입/출력 요청은 새로운 물리 블록 할당을 요구하게 되는 것으로, 볼륨 관리자(300)는 현재의 Last Disk ID 값이 "0"인 점에 착안하여 물리 블록 할당을 디스크 1의 네 번째 블록에 대하여 수행(606-1)하고 입/출력을 수행하는 것이다(단계 913).

<84> 입/출력이 완료된 후, 볼륨 관리자(300)는 Last Disk ID(601)의 변경 필요성을 알려주며, 파일 시스템(100)은 이를 반영하기 위하여 Last Disk ID(601) 값을 "1"로 변경한다(단계 914).

<85> 이에 따라, 물리 블록 할당이 수행되게 되어 해당 파일을 위한 물리 블록의 할당이 전체 디스크에 균등하게 배분되게 되는 것이다.

【발명의 효과】

<86> 상기와 같이 설명한 본 발명은 파일 레벨 스트라이핑 기법을 이용하여 볼륨 관리자(Volume Manager)와 파일 시스템(File System)에서 2개 이상의 디스크(Disk)로 각 파일 데이터(Data)를 분산 저장함으로써, 전체 논리 볼륨뿐만 아니라 각 파일을 위한 데이터 블록들이 논리 볼륨에 참여하는 전체 디스크에 균등하게 분배되어 대용량 파일에 대한 입/출력의 성능을 향상시킬 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

파일 레벨 스트라이핑 장치에 있어서,

물리 블록 넘버(physical block number)를 통해 액세스되며, 정보를 실제로 저장하는 다수의 디스크;

상기 다수의 디스크를 논리적으로 연결하여 하나의 커다란 논리 볼륨을 형성하고, 상기 논리 볼륨을 관리하기 위해 필요한 정보를 다수의 디스크에 기록하고 관리하는 볼륨 관리자;

상기 볼륨 관리자에 의해 제공되는 논리 볼륨을 하나의 저장 장치(storage device)로 인식하며, 상기 제공된 논리 볼륨 상에 파일을 생성한 후, 상기 생성된 파일에 대한 입/출력을 수행하기 위해 논리 볼륨에 대하여 논리 블록 넘버를 적용하는 파일 시스템을 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 장치.

【청구항 2】

제 1 항에 있어서,

상기 논리 볼륨은 볼륨 관리자에 의해 제공되며, 논리 블록 넘버(logical block number)를 통해 액세스 되고, 상기 파일 시스템은 상기 논리 볼륨 상에 파일들을 생성하고, 상기 생성된 파일들을 논리 블록 넘버를 통해 액세스하는 것을 특징으로 하는 파일 레벨 스트라이핑 장치.

【청구항 3】

제 1 항에 있어서,

상기 다수의 디스크중 임의의 디스크 내부 구조는,

상기 볼륨 관리자에 의해 생성되며, 상기 논리 볼륨을 관리하기 위해 필요한 정보를 상기 임의의 디스크에 저장하기 위한 영역인 볼륨 레이블(Volume Label)을 생성하고,

상기 임의의 디스크에 블록 할당을 제어하기 위한 자유공간 비트맵(Free Space Bitmap)을 지정하여 유지 관리하며,

상기 논리 블록이 어떤 물리 블록에 대응되는지에 대한 정보로서 매핑 테이블(Mapping Table)을 유지하며,

상기 임의의 디스크 내에 볼륨 레이블, 자유공간 비트맵, 매핑 테이블 정보 등의 모임인 볼륨 헤더(Volume Header)를 제외한 나머지 공간에 데이터를 저장할 물리 블록들의 모임인 데이터 공간(Data Area)으로 설정 및 관리하는 구조로 이루어진 것을 특징으로 하는 파일 레벨 스트라이핑 장치.

【청구항 4】

파일 시스템과 볼륨 관리자를 구비한 파일 레벨 스트라이핑 기법에 있어서 ,

상기 파일 시스템에서 파일을 생성하는 인터페이스(interface)에 파일 레벨 스트라이핑 지원 여부를 지시하는 옵션(option)을 추가하는 단계;

파일을 위한 아이노드 구조에 Last Disk ID를 확장하는 단계;

상기 파일 시스템에서 상기 파일을 생성할 때 상기 Last Disk ID를 초기화 하는 단계;

상기 파일 시스템에서 파일에 대한 입/출력 요청 시, 상기 볼륨 관리자에 의해 물리 블록 할당이 요구될 때, 상기 Last Disk ID를 기반으로 물리 블록을 할당하는 단계;

상기 볼륨 관리자에 의해 물리 블록 할당이 발생할 때, 파일 시스템에서 이것을 반영하기 위해 상기 Last Disk ID 값을 변경하는 단계를 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【청구항 5】

제 4 항에 있어서,

상기 Last Disk ID 값을 초기화하는 단계에 있어서, 상기 파일 시스템에서 파일을 생성할 때, 옵션으로 주어진 모드(mode)에서 파일 레벨 스트라이핑을 지정하는 비트가 세팅되어 있는지를 판단하는 단계와;

상기 판단 단계에서 해당 비트가 세팅되어 있으면, 상기 파일이 파일 레벨 스트라이핑을 지원한다는 의미로서, 상기 Last Disk ID 값을 논리 볼륨에 참여하는 디스크 수의 범위에 있는 정수 중 무작위 정수를 선택하여 세팅하는 단계;

상기 판단 단계에서 해당 비트가 세팅되어 있지 않으면, 상기 파일이 파일 레벨 스트라이핑을 지원하지 않는다는 의미로서, 상기 Last Disk ID 값을 "-1"로 세팅하는 단계를 더 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【청구항 6】

제 5 항에 있어서,

상기 무작위 정수를 선택하는 이유는 파일 레벨 스트라이핑을 지원하는 모든 파일의 상기 Last Disk ID 초기값이 특정 값으로 고정될 경우, 상기 값에 대응되는 디스크에

많은 데이터 블록의 할당이 집중될 가능성을 제거하기 위한 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【청구항 7】

제 4 항에 있어서,

상기 파일을 생성하기 위한 일반적인 인터페이스에 파일 레벨 스트라이핑을 지원할 것 인지의 여부를 명시하기 위한 옵션은 파일 생성 시 명시되는 옵션인 모드(mode)에 새로운 비트를 추가하여 수행하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【청구항 8】

제 4 항에 있어서,

상기 볼륨 관리자의 물리 블록 할당 방법과 상기 파일 시스템의 상기 Last Disk ID 변경 방법은 사용자가 파일에 대해 입/출력을 요청하는 단계와;

상기 파일 시스템이 요청된 파일 입/출력에 대응되는 논리 블록을 결정하는 단계와;

상기 논리 블록이 결정되면, 하위 입/출력 시스템인 상기 볼륨 관리자에 입/출력을 요청하는 단계와;

상기 볼륨 관리자가 요청된 입/출력을 위해 논리 블록에 대응되는 디스크와 상기 디스크 내의 물리 블록을 결정하기 위하여 주소 매핑 과정을 수행하는 단계와;

상기 주소 매핑 과정의 결과로 디스크와 디스크 내 물리 블록이 결정되면 상기 블록에 대하여 입/출력을 수행하는 단계와;

상기 주소 매핑 결과 논리 블록이 처음 사용되는 것으로 판정되어 물리 블록 할당이 요청되는 경우, 상기 파일에 대응되는 상기 Last Disk ID 값을 체크하는 단계와;

상기 체크 단계에서 상기 Last Disk ID 값이 논리 볼륨에 참여하는 디스크 수의 범위이면, 상기 Last Disk ID 값의 다음 값을 가지는 디스크를 선택하는 단계와;

상기 블록 할당을 수행할 디스크 선택이 결정되면, 상기 결정된 디스크의 자유공간 비트맵을 참조하여 물리 블록 할당을 수행하는 단계와;

상기 물리 블록 할당이 수행된 후, 상기 물리 블록이 할당되었다는 결과를 매핑 테이블에 반영하며, 상기 물리 블록에 대하여 상기 파일을 입/출력하는 단계와;

상기 파일이 물리 블록으로의 입/출력이 완료되면, 상기 Last Disk ID 값을 변경하는 단계와;

상기 변경 단계를 포함하면서 물리 블록의 할당이 전체 디스크에 균등하게 배분하도록 해당 물리 블록에 대하여 입/출력을 반복 수행하는 단계를 더 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【청구항 9】

제 8 항에 있어서,

상기 체크 단계에서 상기 Last Disk ID 값이 "-1"인 경우, 종래의 스트라이핑 방법과 동일하게 전체 논리 볼륨을 대상으로 다음 물리 블록 할당을 수행할 디스크를 결정하기 위한 변수를 참조하여 블록 할당을 수행할 디스크를 선택하는 단계를 더 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

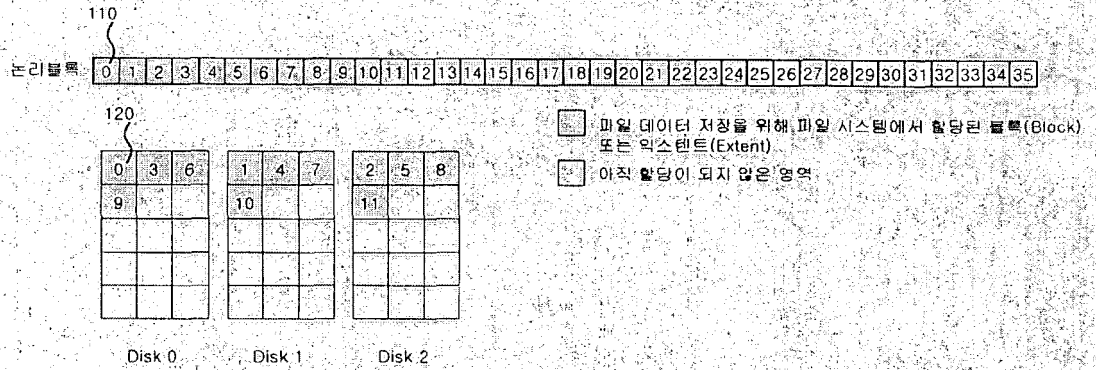
【청구항 10】

제 8 항에 있어서,

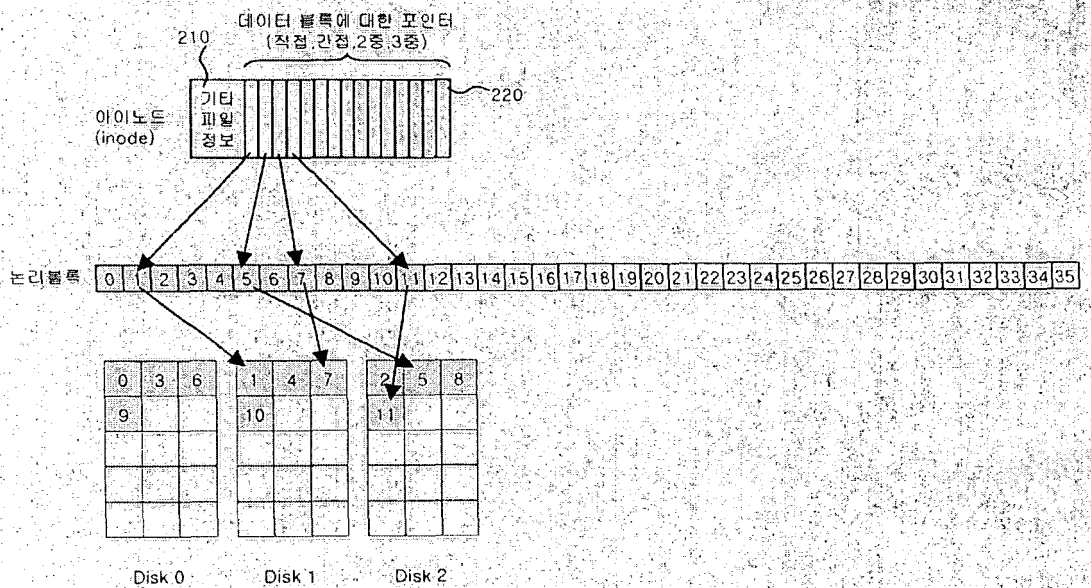
상기 Last Disk ID를 변경하는 단계에서 상기 볼륨 관리자에 의하여 물리 블록 할당이 발생하였고, 상기 파일을 위한 상기 Last Disk ID 값이 "-1"이 아닌 경우에만 상기 Last Disk ID 값을 상기 볼륨 관리자에 의하여 물리 블록 할당이 발생한 디스크의 디스크 아이디(ID)로 설정하는 단계를 더 포함하는 것을 특징으로 하는 파일 레벨 스트라이핑 방법.

【도면】

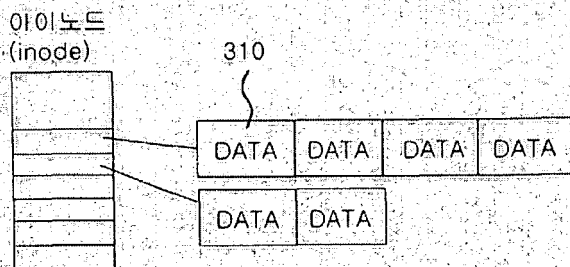
【도 1】



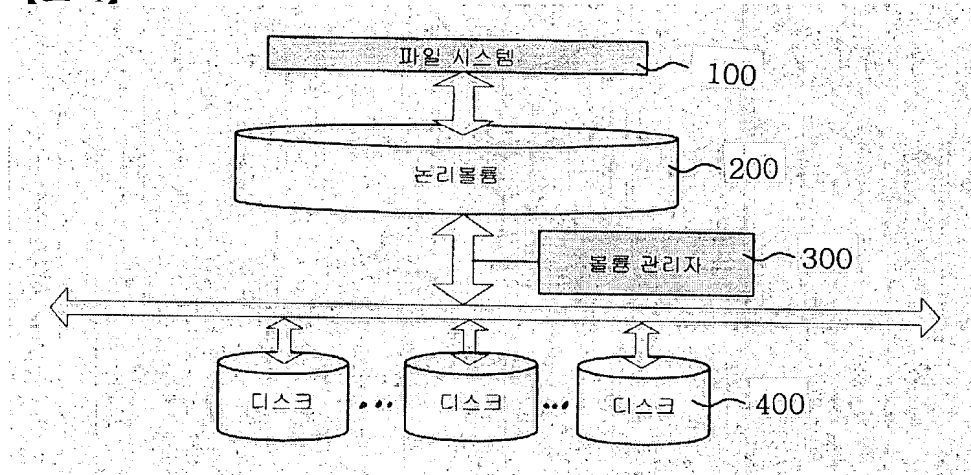
【도 2】



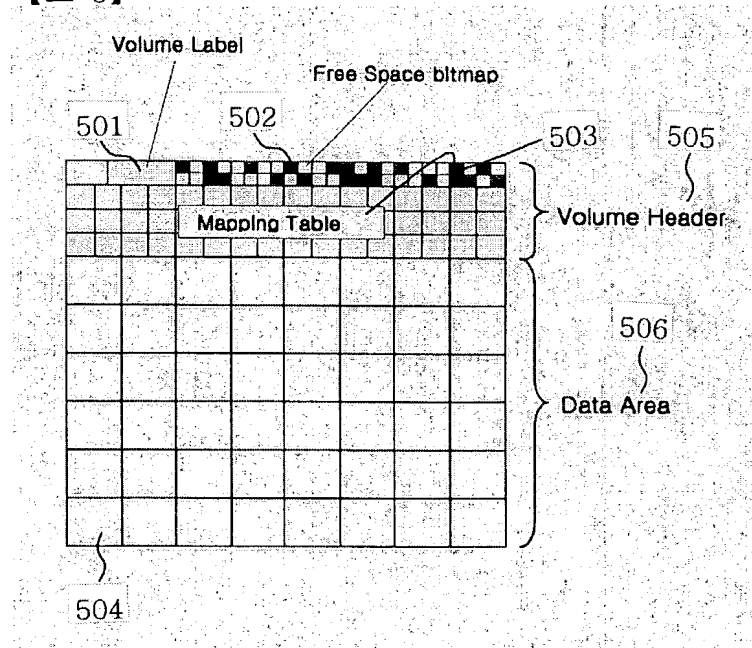
【도 3】



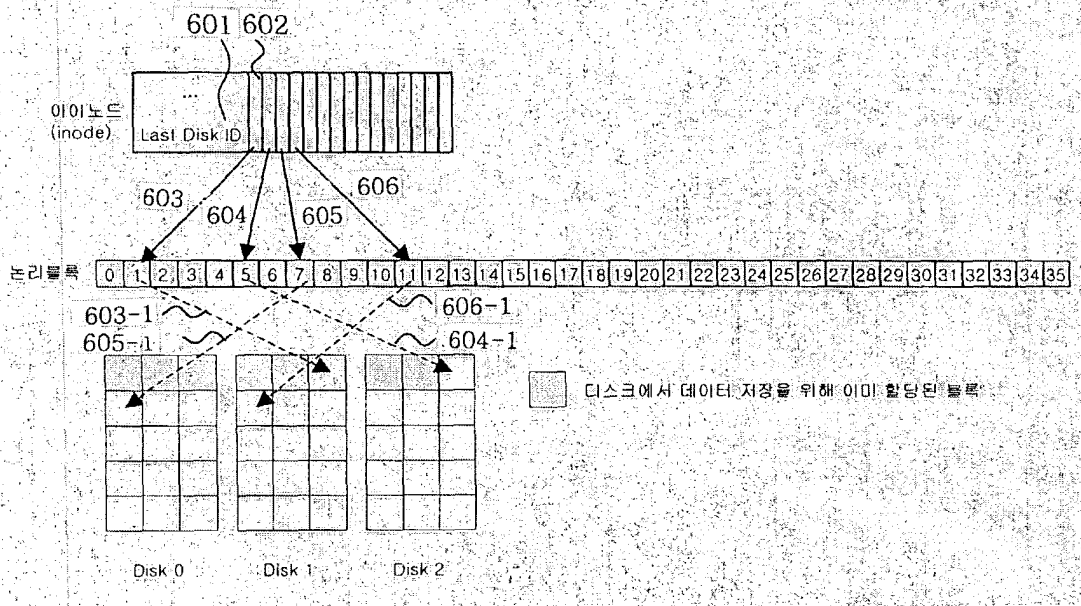
【도 4】



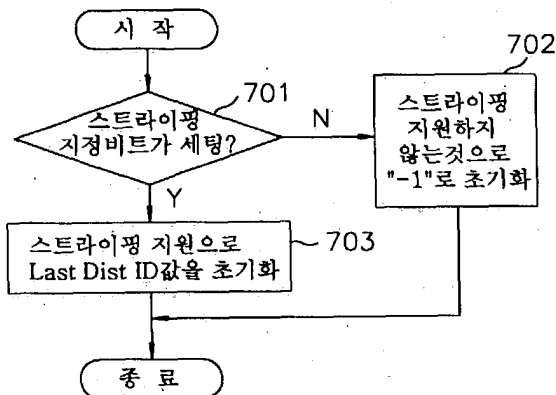
【도 5】



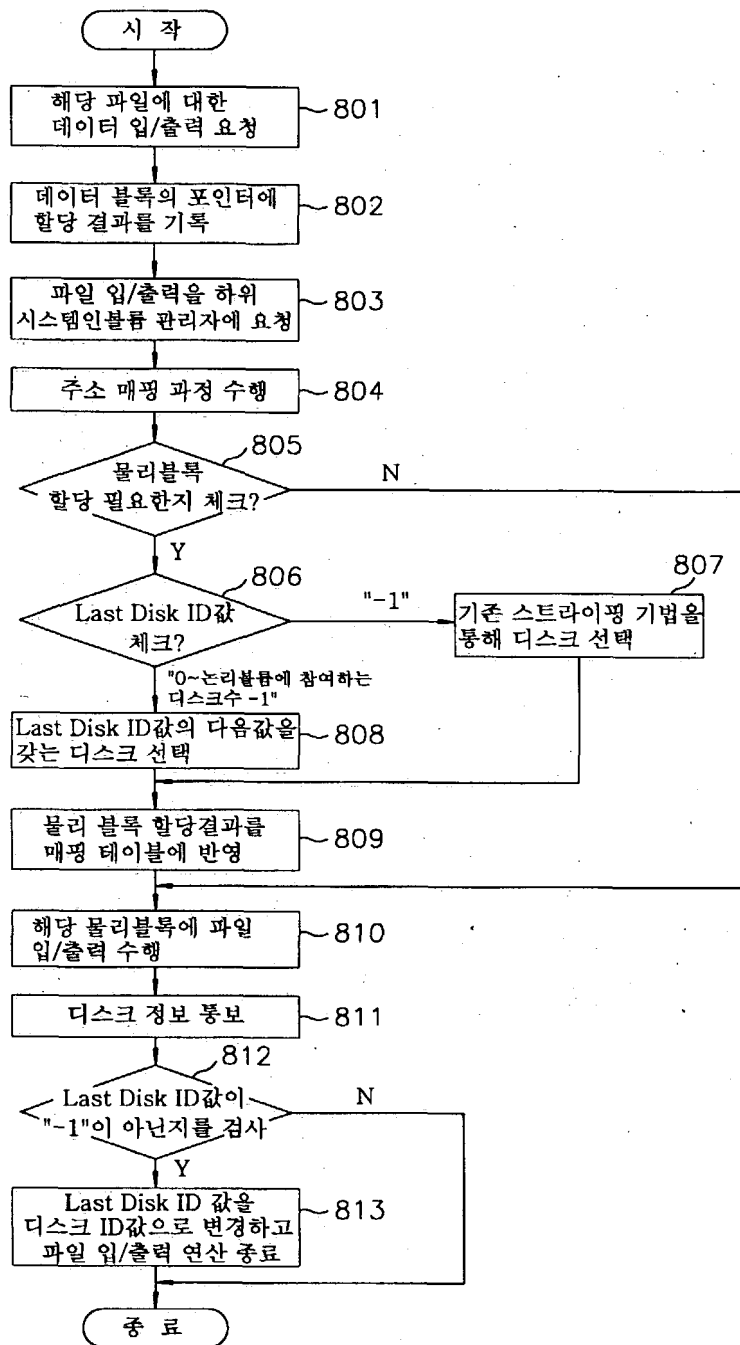
【도 6】



【도 7】



【도 8】



【도 9】

